





z.derying

圖像算法工程師

19人讚同了該文章

原文鏈接：

Find distance from camera to object
using Python and OpenCV

www.pyimagesearch.com



攝像頭測距就是計算照片中的目標物體到相機的距離。可以使用相似三角形（triangle similarity）方法實現，或者使用更複雜但更準確的相機模型的內參來實現這個功能。

使用相似三角形計算物體到相機的距離

假設物體的寬度為W，將其放到離相機距離為D的位置，然後對物體進行拍照。在照片上量出物體的像素寬度P，於是可以得出計算相機焦距F的公式：

$$F = (P \cdot D) / W$$

比如我在相機前24 英寸距離（D=24 inches）的位置橫著放了一張8.5 x 11 英寸（W=11 inches）的紙，拍照後通過圖像處理得出照片上紙的像素寬度P=248 pixels。

所以焦距F 等於：

$$F = (P \cdot D) / W = (248px \cdot 24in) / 11in = 543.45px$$

此時移動相機離物體更近或者更遠，我們可以應用相似三角形得到計算物體到相機的距離的公式：

$$D' = (W \cdot F) / P$$

原理大概就是這樣，接下來使用OpenCV 來實現。

獲取目標輪廓

```
# import the necessary packages
from imutils import paths
import numpy as np
import imutils
import cv2
```

```
def find_marker(image):
```



知乎

首發於

OpenCV 應用系列

```
gray = cv2.GaussianBlur(gray, (5, 5), 0)
edged = cv2.Canny(gray, 35, 125)

# find the contours in the edged image and keep the largest one;
# we'll assume that this is our piece of paper in the image
cnts = cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
c = max(cnts, key = cv2.contourArea)

# compute the bounding box of the of the paper region and return it
return cv2.minAreaRect(c)
```

定義一個 `find_marker` 函數，接收一個參數 `image`，用來找到要計算距離的物體。

這裡我們用一張8.5 x 11 英寸的紙作為目標物體。

第一個任務是在圖片中找到目標物體。

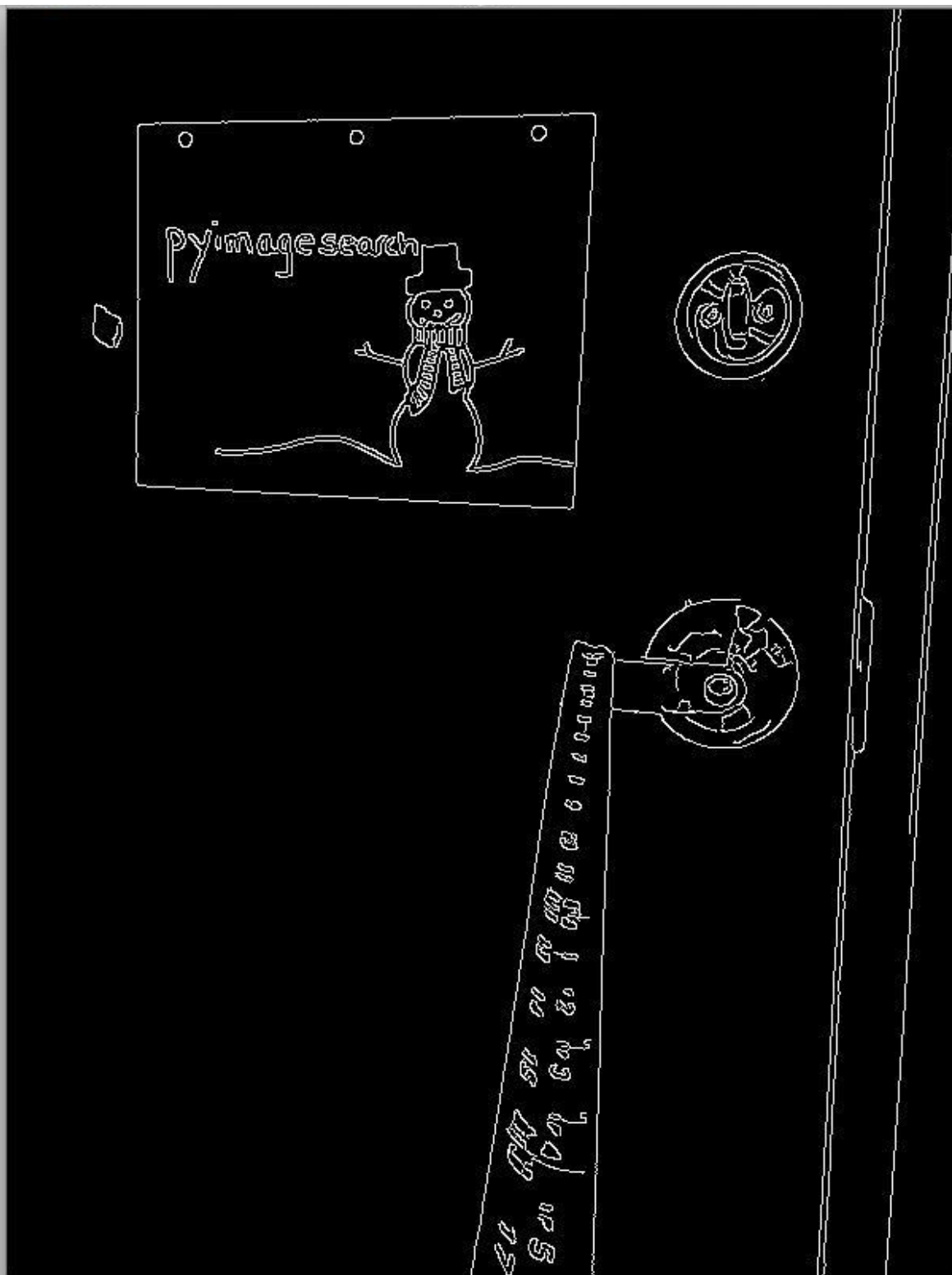
下面這三行是先將圖片轉換為灰度圖，並進行輕微模糊處理以去除高頻噪聲，然後進行邊緣檢測。

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)
edged = cv2.Canny(gray, 35, 125)
```

做了這幾步後圖片看起來是這樣的：



知乎

首發於
OpenCV 應用系列

現在已經可以清晰地看到這張紙的邊緣，接下來需要做的是找出這張紙的輪廓。



```
c = max(cnts, key = cv2.contourArea)
```

用 `cv2.findContours` 函數找到圖片中的眾多輪廓，然後獲取其中面積最大的輪廓，並假設這是目標物體的輪廓。

這種假設只適用於我們這個場景，在實際使用時，在圖片中找出目標物體的方法與應用場景有很大關係。

我們這個場景用簡單的邊緣檢測並找出最大的輪廓就可以了。當然為了使程序更具有魯棒性，也可以用輪廓近似，並剔除不是四個點的輪廓（紙張是一個有四個點的矩形），然後再找出面積最大，具有四個點的輪廓。

注意:關於這個方法，詳情可以查看[這篇文章](#)，用於構建一個移動文本掃描工具。

我們也可以根據**顏色特徵**在圖片中找到目標物體，因為目標物體和背景的顏色有著很明顯的不同。還可以應用關鍵點檢測（keypoint detection），局部不變性描述子（local invariant descriptors）和關鍵點匹配（keypoint matching）來尋找目標。

但是這些方法不在本文的討論範圍內，而且高度依賴具體場景。

我們現在得到目標物體的輪廓了，`find_marker` 函數最後返回的是包含輪廓(x, y)坐標、像素長度和像素寬度的邊框，

計算距離

接下來該使用相似三角形計算目標到相機的距離。

```
def distance_to_camera(knownWidth, focalLength, perWidth):  
    # compute and return the distance from the maker to the camera  
    return (knownWidth * focalLength) / perWidth
```

`distance_to_camera` 函數傳入目標的實際寬度，計算得到的焦距和圖片上目標的像素寬度，就可以通過相似三角形公式計算目標到相機的距離了。

下面是調用 `distance_to_camera` 函數之前的準備：

```
# initialize the known distance from the camera to the object, which  
# in this case is 24 inches  
KNOWN_DISTANCE = 24.0
```



```
# paper is 12 inches wide
KNOWN_WIDTH = 11.0

# Load the first image that contains an object that is KNOWN TO BE 2 feet
# from our camera, then find the paper marker in the image, and initialize
# the focal length
image = cv2.imread("images/2ft.jpg")
marker = find_marker(image)
focalLength = (marker[1][0] * KNOWN_DISTANCE) / KNOWN_WIDTH
```

首先是測量目標物體的寬度，和目標物體到相機的距離，並根據上面介紹的方法計算相機的焦距。

其實這些並不是真正的攝像機標定。真正的攝像機標定包括攝像機的內參，相關知識可以查看[這裡](#)。

使用 `cv2.imread` 函數從磁盤加載圖片，然後通過 `find_marker` 函數得到圖片中目標物體的坐標和長寬信息，最後根據相似三角形計算出相機的焦距。

現在有了相機的焦距，就可以計算目標物體到相機的距離了。

```
# Loop over the images
for imagePath in sorted(paths.list_images("images")):
    # Load the image, find the marker in the image, then compute the
    # distance to the marker from the camera
    image = cv2.imread(imagePath)
    marker = find_marker(image)
    inches = distance_to_camera(KNOWN_WIDTH, focalLength, marker[1][0])

    # draw a bounding box around the image and display it
    box = cv2.cv.BoxPoints(marker) if imutils.is_cv2() else cv2.boxPoints(marker)
    box = np.int0(box)
    cv2.drawContours(image, [box], -1, (0, 255, 0), 2)
    cv2.putText(image, "%.2fft" % (inches / 12),
                (image.shape[1] - 200, image.shape[0] - 20), cv2.FONT_HERSHEY_SIMPLEX,
                2.0, (0, 255, 0), 3)
    cv2.imshow("image", image)
    cv2.waitKey(0)
```

使用for 循環遍歷每個圖片，計算每張圖片中目標對像到相機的距離。

在結果中，我們根據得到的輪廓信息將方框畫了出來，並顯示出了距離。





知乎

首發於
OpenCV 應用系列

總結

通過這篇文章，我們學會了使用相似三角形計算圖片中一個已知物體到相機的距離。



知乎

首發於

OpenCV 應用系列

根據相機的焦距就可以計算圖片中的目標物體到相機的距離。

zxdefying/OpenCV_project

[github.com](#)

編輯於2019-06-02

Python

OpenCV

▲ 贊同19



● 5 條評論

➤ 分享

★ 收藏



文章被以下專欄收錄



OpenCV 應用系列

[進入專欄](#)

推薦閱讀

感興趣區域的移動物體檢測，框出移動物體的輪廓(固定攝像...

感興趣區域、特定區域、框出移動物體的輪廓、越界檢測、入侵物體檢測、使用opencv-python庫的函數cv2.findContours、cv2.approxPolyDP、cv2.arcLength，利用固定攝像頭...
曾伊言



OpenCV



Python



Win

計算機視覺（一）：初識OpenCV

小隱

5 條評論

⇌ 切換為時



知乎

首發於
OpenCV 應用系列

羽織嘎

2019-10-06

照片上的像素寬度是直接尺子測量得到的麼？

贊



hoppss

2019-10-16

有什麼簡單方法獲取，像素物理尺寸和焦距

贊



jaypeng

2019-10-24

樓主有試過攝像頭不垂直的情況嗎

贊



知乎用戶 回复 jaypeng

01-14

有人試過嘛？

贊



知乎用戶

03-05

怎麼都是光頭博士哥哥的blog翻譯啊

贊

